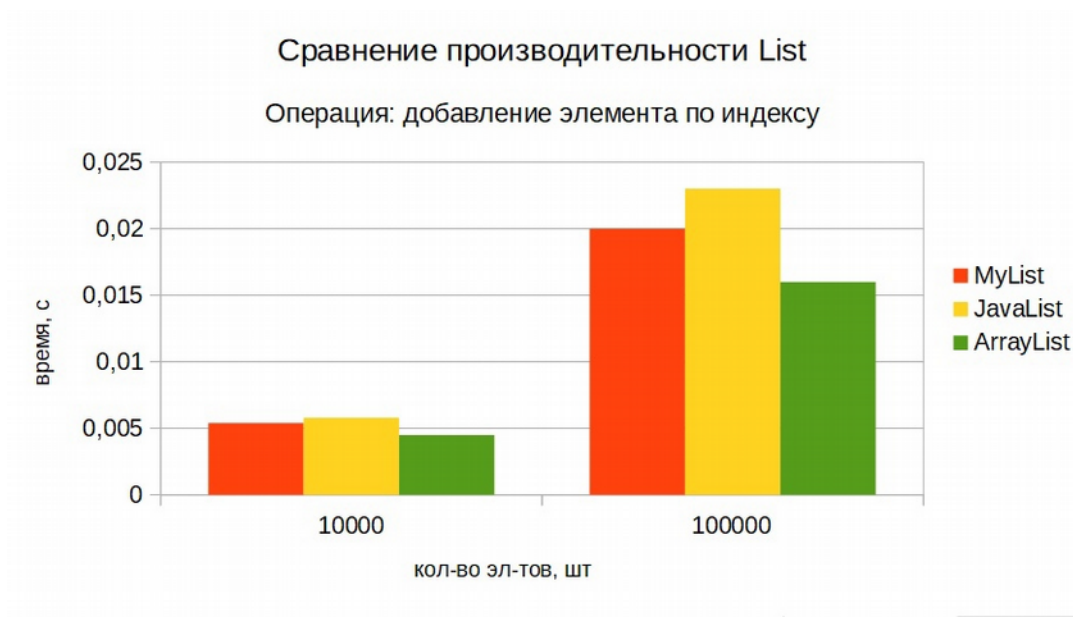


## Сравнение производительности коллекций для основных операций

Используя System.nanoTime оценим примерно время выполнения основных операций.

### 1. ArrayList, LinkedList, MyLinkedList





Подытоживая полученные данные, имеем следующее:

LinkedList проигрывает ArrayList(так как индекс выбирается случайным образом и данных достаточно много).

Когда использовать LinkedList:

1. Необходимо много данных добавлять в начало и конец списка.
2. Удалять, извлекать с начала и с конца списка.

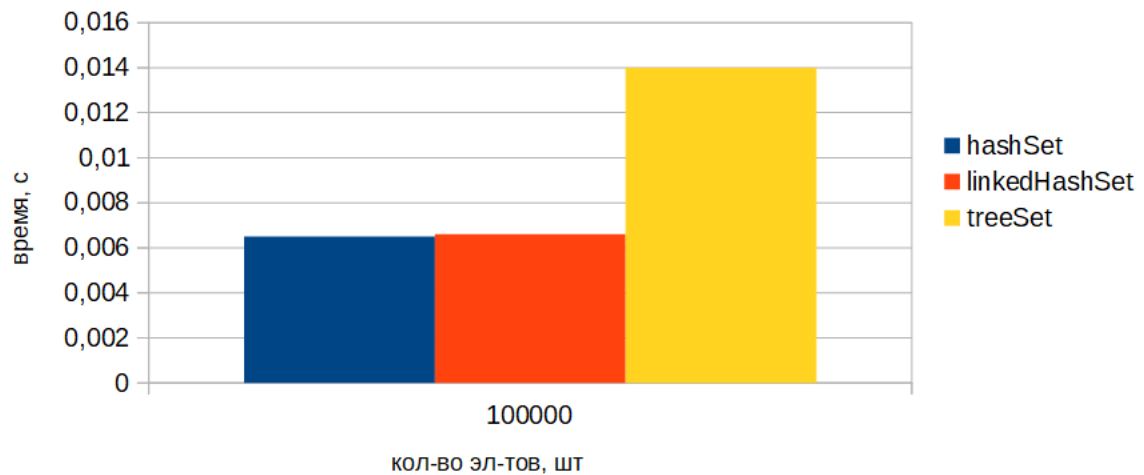
Когда использовать ArrayList:

1. Получить элемент по индексу.
2. Добавить элемент в конец.
3. Удалить элемент (кроме начала списка).

## 2. HashSet, LinkedHashSet, TreeSet

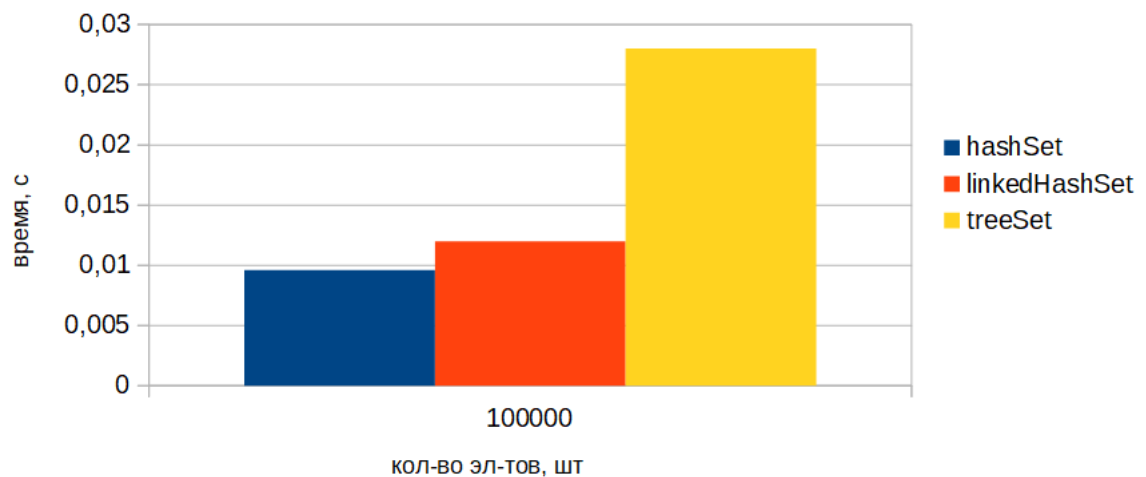
### Сравнение производительности Set

Операция: добавление элемента



### Сравнение производительности Set

Операция: удаление элемента

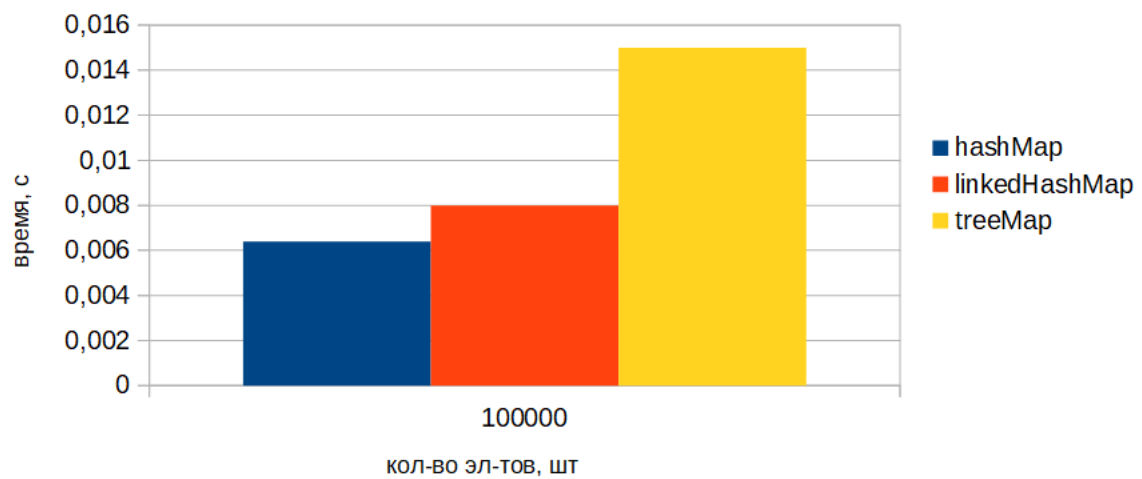


Set представляет собой множество без дублирующих элементов. Если нам нужна быстрая коллекция, используем HashSet, если отсортированная коллекция, то TreeSet, если нужна коллекция, которая может хранить порядок вставки, используем LinkedHashSet.

### 3. HashMap, LinkedHashMap, TreeMap

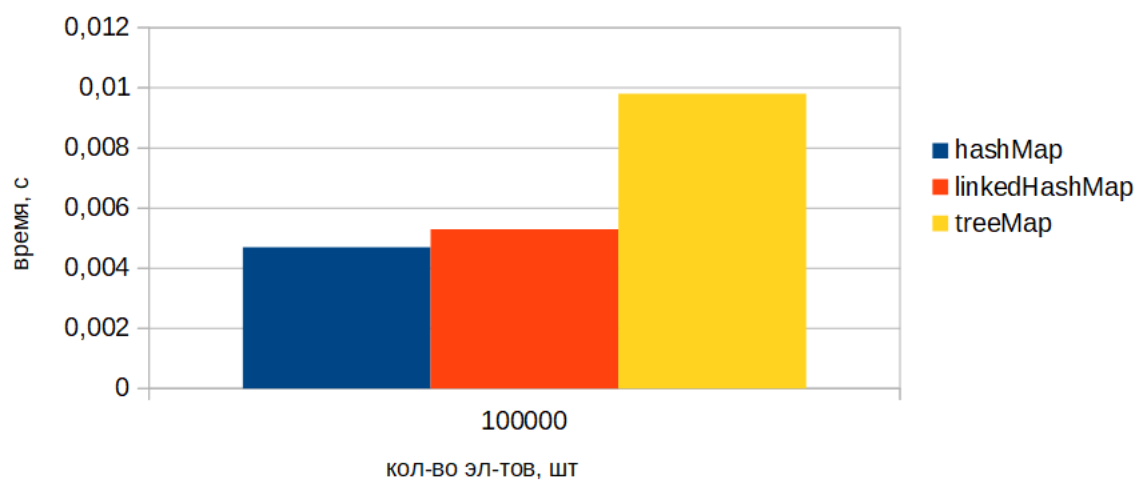
#### Сравнение производительности Map

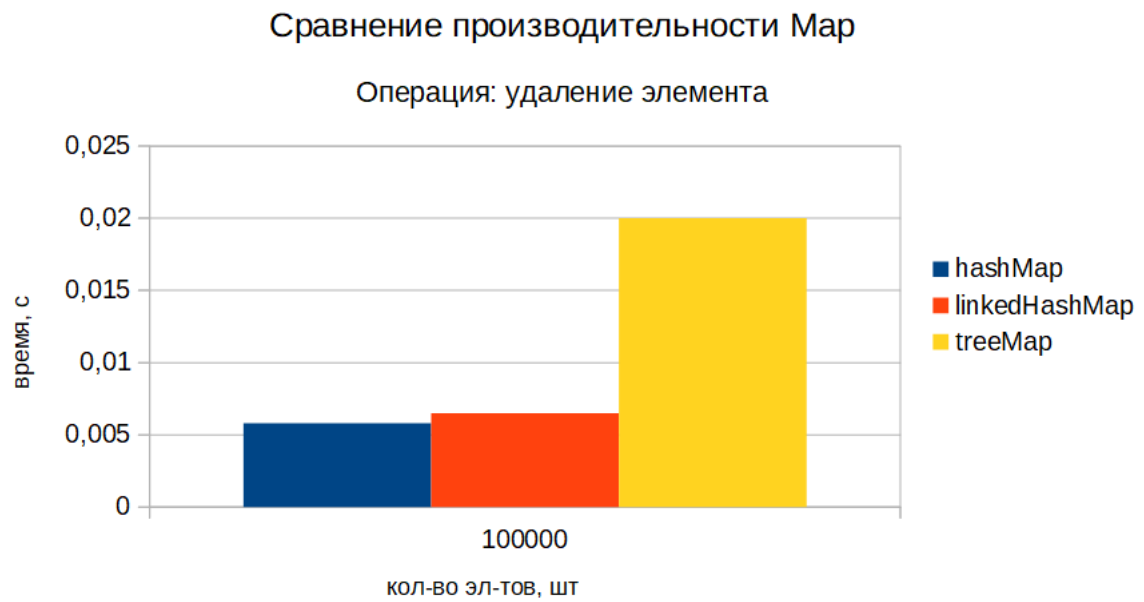
Операция: вставка элемента



#### Сравнение производительности Map

Операция: получение элемента





Map – структура, которая хранит пару (ключ, значение). Выбор аналогично структурам Set.